NAME

x10aux - Auxiliary input to Heyu via RF

DESCRIPTION

Heyu is a program for controlling an X-10 "CM11A" home control device. See the *heyu*(1) man page for usage information.

This page contains information about the capability of Heyu to receive and process signals from RF remotes and sensors using a WGL W800RF32A, an X-10 MR26A, or an RFXCOM X10 RF receiver connected to a second serial port.

HARDWARE

The W800RF32A is manufactured by WGL & Associates (http://www.wgldesigns.com). Two models are available - the US/Canada model operates at a frequency of 310 MHz and the International model (W800RF32AE) at 433.92 MHz. It is capable of receiving signals from standard X10 RF remotes and sensors like the X-10 HR12A "PalmPad", from security X10 remotes and sensors like the X-10 DS10A Door/Window Sensor, and from older entertainment X-10 remotes like the UR81A Universal Remote which are designed to be used in conjunction with an X-10 MR26A Receiver. Its receiving range is excellent.

The X-10 MR26A is capable of receiving standard X10 and the older entertainment X10 signals, but not the security X10 signals. Its receiving range is somewhat limited, perhaps 20-30 feet.

The RFXCOM X10 receiver (http://www.rfxcom.com) is available in a US/Canada 310MHz version, an International 433.92 MHz version, and a dual-frequency version. All versions receive signals from standard, entertainment, and security RF remotes and sensors (which transmit at their frequency). The 433.92 MHz and dual-frequency versions can additionally receive signals from various other sensors like Oregon Temperature/Humidity/Barometric Pressure sensors.

The RFXCOM X10 receiver is supported by Heyu in both variable length packet and 32 bit (W800 emulation) modes.

The RFXCOM is a USB device but has a built-in FTDI USB-to-Serial converter and communication with it is the same as with a serial port (assuming your OS supports the FTDI chipset, as does Linux).

All of these devices are strictly RF receivers and have no transmitting capability.

QUICK START

Stop Heyu (if already running) with the command 'heyu stop'. Plug a W800RF32A/AE or MR26A receiver into a serial port. Or plug an RFXCOM receiver into a USB port and determine the serial device assigned by the OS. (Under Linux, the first USB device plugged in will normally be assigned to /dev/ttyUSB0, or possibly to /dev/usb/ttyUSB0, the second USB device to /dev/ttyUSB1, etc.)

Describe for Heyu the port and receiver being used with the following directive in the Heyu configuration file:

TTY_AUX <serial port> <receiver device>

Example:

TTY_AUX /dev/ttyS1 W800RF32A

The configuration directives TRANSCEIVE and RFFORWARD determine whether signals from a Standard X10 transmitter (like the HR12A PalmPad) are directly transceived to the powerline or are forwarded to the Heyu Engine for processing. The defaults for these directives are "TRANSCEIVE ALL" and "RFFORWARD NONE".

Run ´heyu start´, and in another xterm ´heyu monitor´. Verify that the RF signals are being correctly received. Transceived signals will appear in the monitor window with source "snda", indicating they were sent by the auxilliary daemon heyu_aux.

For transmissions from Security X10 sensors like the X10 DS10A Door/Window sensor (with an RF receiver capable of receiving these transmissions) the Heyu monitor will display an "RFdata" signal.

Example:
... rcva func RFdata : Type Sec ID 0x23 Data 0x04

Before Heyu can decode the signal data it has to know the type of sensor. This is accomplished by mapping the sensor module type and its ID (0x23 in this example) to an otherwise unused housecode|unit address with an ALIAS directive in the configuration file, e.g.,
ALIAS Back_Door B2 DS10A 0x23

Then after running ´heyu restart´, opening the door will result in a monitor display like:
... rcva func Alert : hu B2 swMin (Back_Door)
where the flag swMin indicates the "Delay" slider switch on the DS10A is set to the Min position. The "rcva" means the signal was received from the Heyu Auxilliary daemon, heyu_aux. (This signal source will have to be specified in the SCRIPT launch conditions if the alert signal is to launch a script.)

RF TRANSMITTERS

In addition to the various standard X10 remotes and sensors, Heyu currently includes RF decoder modules for the following RF security and entertainment remotes and sensors:

North American models:
SH624 Security Remote
KR10A Security Keychain Remote
DS10A Security Door/Window Sensor
MS10A Security Motion Sensor (See section MS10A WARNING).
UR81A Universal Remote (Entertainment).

International models:
Marmitek DS90 Security Door/Window Sensor (See section "SPECIAL DS90/DS18-1 SETUP" towards the bottom of this man page.)
ElekHomica DS18-1 Door/Window Sensor (2 channel. Equivalent to Marmitek DS90)
ElekHomica DS18 Door/Window Sensor (1 channel, 433.92 MHz version of DS10A)
Marmitek SD90 and SD10 Smoke Detectors (See section "SPECIAL SD90 SETUP" towards the bottom of this man page.)
Marmitek MS90 Security Motion Sensor
ElekHomica EH-CWSD10 Smoke Detector
ElekHomica EH-WD210 Water Detector
Marmitek GB10 Glass Break Detector
(Also sends a sDusk signal at dark)

Decoders are also included for RFXSensor and RFXMeter signals, and signals from the DigiMax 210 Thermostat and Oregon sensors. Owners of these devices should see man pages x10rfxsensors(5), x10rfxmeters(5), x10digimax(5), and/or x10oregon(5).

Modules for other remotes and sensors will be added once the RF code generated by each button-press or sensor action is known. (Heyu displays this information.)

Security remote and sensor devices transmit two significant bytes of code at each button-press or sensor action. The first of these is an identification byte which is (nominally) unique for each individual device; the second describes the function of the particular button-press or sensor action.

Older entertainment remotes like the UR81A transmit two significant bytes of code. The first of these is either constant or restricted to a few values; the second describes the function of the button-press.

The way each of the bytes are encoded for RF transmission allows distinguishing between standard, security, and entertainment codes.

MODULE OPTIONS

REVERSE keyword.
The Alert/Clear action of security Door/Window sensors may be swapped by including the keyword REVERSE as a parameter to the ALIAS directive which maps the sensor ID to a Housecode|Unit address. With this option the Alert signal is issued when the door/window is closed and the Clear signal when it is opened. This option is currently supported by the models DS10A and DS90 sensors. It was added so that these sensors can be used with a N/O switch instead of the N/C magnetic switch supplied with the unit.

MAIN keyword
AUX keyword
These keywords are currently supported only by the DS90 Security Door/Window sensor. See the special setup instructions for this sensor in the SPECIAL DS90 SETUP section toward the bottom of this man page.

BASIC OPERATION

In order to receive RF signals, Heyu relies on the heyu_aux daemon, which is started either manually with the ´heyu aux´ command or automatically in the startup sequence with the ´heyu start´ command. The serial port and attached receiver device must be specified in the Heyu configuration file with the TTY_AUX directive. The syntax for this directive is:
TTY_AUX <serial port> <receiver device>
where <receiver device> is W800RF32A, MR26A, or RFXCOM. Example:
TTY_AUX /dev/ttyS1 W800RF32A

RFXCOM defaults to the variable length packet mode model, RFXCOMVL. The 32 bit W800 emulation mode RFXCOM32 may be specified if necessary.

There is no default for this directive.

Standard X10 RF signals received by heyu_aux may either be directly transceived to X10 powerline code or may be forwarded to the heyu_engine and used to launch scripts without the delay inherent in X10 powerline communication. The alternatives are controlled by the two configuration directives, TRANSCEIVE and RFFORWARD. The syntax for these directives is:
TRANSCEIVE <list>
RFFORWARD <list>
where <list> may be the keywords ALL or NONE, or may be a string of housecode enclosed in square [] brackets.
Example:
TRANSCEIVE [BFH]
RFFORWARD [IJK]
which will transceive RF signals on housecode B, F, and H, and forward RF signals on housecode I, J, and K. RF signals on all other housecodes will be ignored.

Either of these directives may also use the keyword ALLEXCEPT followed by the square bracketed housecode list to include all housecodes except those in the list.
Example:
TRANSCEIVE [BFH]
RFFORWARD ALLEXCEPT [BFHLM]
In this example, housecodes B, F, and H will be transceived, housecodes L and M will be ignored, and all others will be forwarded.

Any given housecode may not be both transceived and forwarded.

The default for the TRANSCEIVE directive is ALL, and that for the RFFORWARD directive is NONE.

Finer grained control is available from special module types used in an ALIAS directive which can override the TRANSCEIVE and RFFORWARD selections for specific units and functions within a housecode. These module types are:
PALMPAD (or HR12A) - Controls RF On, Off, Dim, and Bright
KEYCHAIN (or KC624) - Controls RF On and Off
ONLYON - Controls RF On
ONLYOFF - Controls RF Off
MS12, MS13, MS14, MS16 - Controls RF On and Off.

The MSxx module types differ from the KEYCHAIN module type only in that they are defined as "sensors" and will be listed in the table displayed by ´heyu show sensors´.

Each of these special module types requires one of the parameters TRANSCEIVE, RFFORWARD, or RFIGNORE to define its functionality.
Example:
ALIAS XMMS_Control D1-4 PALMPAD RFFORWARD
which would direct heyu_aux to forward On/Off/Dim/Bright signals from an X-10 PalmPad (or any other RF remote) on housecode D, units 1 through 4, regardless of the selections in TRANSCODE and RFFORWARD (which will otherwise control other RF signals on this housecode).

Example:
ALIAS LightIgnore B2 KEYCHAIN RFIGNORE
would direct heyu_aux to ignore RF signals from the light sensor on Address+1 of a (non-security) Motion Sensor, e.g., the X-10 MS14A, set to address B1 (which often causes collision problems when the sensor´s "motion" signal turns on a lamp within view of the sensor).

If, for whatever reason, you have an external transceiver like a TM751 or RR501 in operation, Heyu should usually not transceive on the same housecode lest there be signal collisions on the AC power line.

Note: Heyu identifies signals transceived by heyu_aux as having the source SNDA. Signals forwarded to heyu_engine are identified as having source RCVA. Remember this when using these signals to launch a script.

Security and Entertainment X10 RF signals received by heyu_aux are decoded and processed by the Heyu State Engine daemon, heyu_engine. Since these types of signals contain no Housecode/Unit identification, the transmitting device must be mapped to a Housecode and Unit in an ALIAS configuration directive for the RF signal to be decoded by the Heyu Engine. Once decoded, the signals can be used to launch scripts or control various Heyu features like a home security system.

Heyu identifies security and entertainment signals from heyu_aux as originating from source RCVA. Remember this when using these signals to launch a script.

For security devices, the identification of the individual device (or devices if you have more than one of the same type) must be provided with the ALIAS directive. The syntax is:
ALIAS <label> <housecode|unit> <device model> <ID> [<ID> [<ID>...]]
where <ID> represents the security ID of a device expressed as a hexadecimal number, either with or without the "0x" prefix. Up to 16 security IDs can be associated with a single housecode|unit address for security remotes.

Note: multiple device IDs are normally mapped to a single housecode|unit address only for security remotes of the same model. Security sensors must be mapped to different addresses so the signals from each can be distinguished. Examples:
ALIAS my_sh624_remote D12 SH624 0x1c b2
ALIAS back_door C3 DS10A 0x65

To determine the security ID of a device, start Heyu normally and open a Heyu Monitor window. Operate the device(s) in question by pressing a button, opening the door, or whatever it takes to make it send an RF signal. Heyu will display the raw RF signal in the monitor window like this:
rcva func RFdata : Type Sec ID 0x65 Data 0x04
which provides the information that a signal was received by heyu_aux (rcva) and that it is from a device of

type Sec[urity] with ID 0x65. Once we have added the ALIAS directive (in the back_door DS10A example above) to the configuration file and restarted Heyu, the same signal now interpreted by heyu_engine will be displayed in the monitor as:

```
rcva func      Alert : hc C unit 3 swMin (back_door)
```

Indicating the door was opened and that the DS10A has its slider switch set to the "min" position.

Most X10 Security devices actually send a 16-bit ID code, however the upper byte is received only with an RFXCOM receiver in variable length packet mode. The examples here illustrate only the 8-bit code which would be received by a W800RF32A/AE receiver or RFXCOM in 32 bit mode. In the ALIAS directive, use whatever ID code, 16-bit or 8-bit, is reported by Heyu from your RF receiver.

For entertainment remotes like the UR81A, the ID doesn´t change. It is built into the model and isn´t specified in the ALIAS. So using the UR81A as an example, we could use the directive:

```
ALIAS my_ur81a  B2  UR81A
```

The RF signals from entertainment remotes are currently decoded by heyu_engine only as virtual data (´vdata´) signals. Heyu scripts can examine the data value (environment variable X10_Vdata) to determine what action to take for a particular button-press. An example script is UR81A_Action.sh found in the Utilities section of the Heyu website (http://www.heyu.org).

SECURITY FUNCTIONS AND FLAGS

The "Arm" and "Disarm" RF signals from security remotes like the X-10 SH624 and KR10A correspond to functions "arm" and "disarm". They control Heyu´s global security flags ("armed", "disarmed", "armpending", "home", and "away") the same as if the corresponding ´heyu arm ...´ or ´heyu disarm ...´ commands were entered from the keyboard. (Global flags may be tested in the launch conditions for any script.)

Signals from security remotes and sensors also set local flags for the actual or implied switches on the devices: "swmin", "swmax", "swhome", "swaway", and finally "lobat" for a sensor low-battery flag. (Local flags may only be tested in a launch condition based on a signal received from the particular device which set that flag.)

Security sensors send the RF signals "Alert" or "Clear", corresponding to functions "alert" and "clear". They periodically repeat the current state of the device in a signal approximately every 60-90 minutes, just to let the host system (Heyu in this case) know they are functioning normally.

Don’t confuse the functions "arm" and "disarm" with the flags "armed" and "disarmed", and don’t confuse the local flags "swhome" and "swaway" with the global flags "home" and "away".

CONFIGURATION DIRECTIVES

The TTY_AUX, TRANSCEIVE, RFFORWARD, and ALIAS directives are described earlier in this document.

TRANS_DIMLEVEL directive

This directive specifies the dim level for each RF Dim or Bright signal transceived by heyu_aux. This is the same level as would be sent with the ´heyu dim ...´ or ´heyu bright ...´ command from the keyboard. The default value is 2, which produces a change of about 6 percent in brightness. Setting the value to 3 would produce a change of about 11 percent. The allowed range for this directive is 1-22, the same as for commands sent from the keyboard. Example:

```
TRANS_DIMLEVEL  2
```

AUX_REPCOUNTS directive

RF transmitters of all types generally repeat the transmission in multiple bursts. For example the X-10 HR12A "PalmPad" transmits a minimum of 6 bursts - more if button is held down; the X-10 security remotes and sensors typically transmit 5-7 bursts. This directive instructs heyu_aux how to handle multiple bursts in an uninterrupted sequence by providing 3 numbers:

```
AUX_REPCOUNTS  <MIN> <REPEAT> <MAX>
```

where:

<MIN> is the minimum number of identical RF bursts in a row
required for heyu_aux to issue its first response, i.e.,
transceive the signal or send the signal to heyu_engine.
(Default is 1 for the W800RF32A and RFXCOM, or 2 for the MR26A,
which is more susceptable to noise.)

<REPEAT> is the number of identical RF bursts in a row before
heyu_aux will issue additional responses. (Default 8)
If <REPEAT> is set to zero, no more than the first response
will be issued. Otherwise, setting the value of <REPEAT> too
low can result in overruns - RF signals will accumulate
in the system´s serial driver buffer faster than they
can be transceived.

<MAX> is the number of bursts in a row without any break at which
point heyu_aux will stop issuing its normal responses and
instead issue a "RF Flood : Started" signal. (Default 200)
Once there´s a break in the flood, heyu_aux will issue a
"RF Flood : Ended" signal.
If <MAX> is set to zero, heyu_aux will continue to send responses
without limit and there will be no "RF Flood" signals.

The purpose of the <MAX> count is to protect the system from being overwhelmed by an accidental (or deliberate) unbroken flood of RF bursts, e.g., from a stuck button on a remote. Once there's an interruption in the flood, the counting reverts back to <MIN>. Heyu can be configured to launch a "-rfflood" script when it receives a RF Flood Started or Ended signal.

Most users won't need to change the defaults for this directive.
Example:

```
AUX_REPCOUNTS 1  8  200
```

will result in a signal being transceived or sent to the heyu_engine on the 1st, 9th, 17th, ..., 193rd burst. Then RF Flood messages will be sent on the 200th, 400th, 800th, etc., burst.

SUPPRESS_RFXJAM directive
Older firmware versions of the RFXCOM receiver sent a special signal when they detected RF jamming, however the system was prone to many false positives and the feature was removed.
The options for this directive are YES or NO, with the default being NO. With this default, jamming signals from the older RFXCOM receivers are reported in the Heyu Monitor and Log file as "RF Jamming : Started|Ended Main|Aux", where Main and Aux refer to the RFXCOM Master and Slave receivers. If set to YES, the jamming signals are treated as RF Noise.

HIDE_UNCHANGED directive
This directive allows the display of signals in the Heyu Monitor and log file to be suppressed if successive signals are unchanged, for example the periodic "heatbeat" signals from security sensors or temperature signals from temperature sensors.
With the default value of NO for this directive, the log file and monitor will be cluttered with between about 16 to 24 superfluous (typically "Clear") signals daily for each security sensor, or far more from sensors like Oregon temperature sensors which transmit approximately every 30 to 90 seconds.

If the value of this directive is set to YES, then the signal will be displayed in the monitor and log file only when it represents a change from the previous state, or if the signal launches a script. Only the display is hidden - the processing by heyu_engine continues normally.

DISPLAY_RAW_RF directive
This directive instructs Heyu whether or not to display the raw RF data bytes from the receiver device. The choices are the default "NONE" to not display any raw data, "NOISE" to display data which heyu_aux judges to be RF noise, or "ALL" to display both noise and normal raw RF data.

The display of raw data is in addition to the normal decoded data display. Displaying raw data requires writing a _lot_ of data to the spool file which can interfere with CM11A communications, so this directive should be left at the default "NONE" (or "NOISE") except for testing and debugging (or just to see what it looks like).

Note: Some versions of the W800RF32A are said to receive 4-byte RF data from newer X10 entertainment remotes like the CR14A "Pan ’n Tilt" remote and the UR89A "Lola" remote. With the current absence of models for these remotes in Heyu, heyu_aux is forced to classify RF data which might be received from these remotes as RF noise.

SECURID_16 directive
Is used with the RFXCOM receiver in variable length mode to instruct Heyu how to handle 16-bit Security IDs. The default is YES, to use 16-bit IDs. If set to NO, Heyu will mask off the upper byte and use only the lower byte, which corresponds to the 8-bit ID used by the W800RF32 and RFXCOM receiver in 32 bit mode. This directive is provided primarily for those who have configured a large number of sensors using the 8-bit IDs, until they have a chance to reconfigure them.

SECURID_PARITY directive
Some security sensors appear to have a firmware bug whereby a parity bit for the upper byte of a 16-bit ID isn´t set properly. With the default value of YES for this directive, the signal will be classified as NOISE and ignored. Setting the value of this directive to NO instructs Heyu to ignore this parity bit, which is less risky than ignoring the signal.

LAUNCHING SCRIPTS

In addition to the standard X10 functions transceived by heyu_aux (with source SNDA), the following functions received by heyu_engine (with source RCVA) from heyu_aux are available for launching scripts: "arm", "disarm", "panic" "alert", "clear", "slightson", "slightsoff", "vdata", "test", and "tamper".

Other special functions which can launch scripts are described in the Heyu man pages for RFXSensors, RFXMeters, Digimax, and Oregon sensors.

Don´t forget to include the source keyword(s) in the launch conditions.

The keywords and flags which can be tested in the launch conditions for a script in addition to the usual keyword "changed" and the common flags 1-16 are: "armed", "armpending", "disarmed", "home", "away", "swhome", "swaway", "swmin", "swmax", "lobat", "tamper", "main", and "aux". (The last three currently relate only to the DS90 Security Door/Window sensor.)
Example:

```
ALIAS  side_window  E7 DS10A  0x3d

SCRIPT side_window alert armed away rcva :: heyu turn siren on
```

The special launcher type "-rfflood" will launch a script when an RF Flood signal is received. The flags that can be tested in the launch conditions for this launcher are the special flags "started" or "ended", the common flags 1-16, and the security flags "armed", "armpending", "disarmed", "home", and "away".
Example:

```
SCRIPT -rfflood started armed away :: heyu on siren
SCRIPT -rfflood ended :: heyu off siren
```

MS10A WARNING

When the total voltage of the four AA cells in the MS10A falls below about 4.3 Volts, THE SENSOR WILL NO LONGER DETECT MOTION. Its heartbeat signal then is always Alert with the LoBat flag, which continues to be transmitted until the battery voltage is somewhat lower. To avoid false alarms, Heyu scripts should always check for the Alert/LoBat condition before checking for Alert alone, e.g.,

```
SCRIPT MyMS10A alert lobat rcva :: echo "Low battery" | mail
SCRIPT MyMS10A alert rcva :: call_police.sh
```

SPECIAL DS90/DS18-1 SETUP

The DS90 and DS18E Security Door/Window Sensors have two independent circuits. In addition to the main circuit which is actuated by the magnetic door/window switch, there is an auxiliary circuit actuated by connecting a switch to a pair of internal contacts.

The DS90/DS18-1 also has a "tamper" switch actuated by removing the cover from the unit which will issue a "Tamper" command and set the Heyu tamper flag. (The tamper flag is sticky and must be cleared by executing a ´heyu clrtamper´ command.)

Each circuit has its own security ID. The security IDs are related by the following formula, so given one the other can be derived:
(bit-reversed)AUX_ID = 1 + (bit-reversed) MAIN_ID.

This sensor can be configured in Heyu several different ways:

Map the main and aux circuits to different housecode|unit addresses. Simply use the main ID in one ALIAS directive and the aux ID in another ALIAS directive.
Example:

```
ALIAS kitchen_door  C12  DS90  0x63
ALIAS kitchen_deadbolt C13 DS90 0xE3
```

Map both main and aux circuits to the same housecode|unit address by including both security IDs in the one ALIAS directive. The signals from each will be distinguished by flags MAIN or AUX.
Example:

```
ALIAS kitchen_entry  C12  DS90  0x63  0xE3
```

Note: A potential hazard with mapping both circuits to the same housecode|unit address is that they both use the same activity timer. So the failure of one circuit won't be tagged as "inactive" so long as the other circuit is still working.

If both circuits are mapped to the same address, the raw data from the AUX sensor is stored in the "memory level" byte in the state table and can be recovered with ´heyu rawmemlevel Hu´.

Use only one of the two circuits and ignore signals from the other. To do this, include the security ID for whichever circuit you want to use in the ALIAS directive. Tell Heyu which one it is by adding the keyword either MAIN or AUX as a parameter to the directive.
Example:

```
ALIAS kitchen_door  C12  DS90  0x63  MAIN
```

In the above, Heyu will compute the AUX ID (0xE3) and ignore signals received from it.

SPECIAL SD90 SETUP

The Marmitek SD90 Smoke Detector transmits signals at two independent ID addresses, an "Emergency" or "Test" address and a "Sensor" address.

Marmitek security base stations apparently use only the signal at the Emergency address, and with the factory default SD90 setting the signals at the Sensor address are disabled. This is unfortunate because the Sensor transmissions include two important features which are absent in the Emergency transmissions: a periodic "heartbeat" signal and a low battery flag.

The Emergency and Sensor addresses may individually be programmed to a value 1 through 16. The following table displays the (8-bit) security ID for each programmed address.

Note: An RFXCOM RF receiver in the default variable length mode will display a 16-bit security ID with a high byte of 0x54 and a low byte as shown in this table, e.g., 0x54C0 for Emergency address 1.

Address	Emergency	Sensor

1	0xC0	Disabled (Factory setting)
2	0xC1	0xD1
3	0xC2	0xD2
4	0xC3	0xD3
5	0xC4	0xD4
6	0xC5	0xD5
7	0xC6	0xD6
8	0xC7	0xD7
9	0xC8	0xD8
10	0xC9	0xD9
11	0xCA	0xDA
12	0xCB	0xDB
13	0xCC	0xDC
14	0xCD	0xDD
15	0xCE	0xDE
16	Disabled	0xDF

Each installed SD90 Smoke Detector unit should be set to its own unique addresses. It´s probably a good idea to check with nearby neighbors who may have a SD90 within range of your RF receiver.

While the Emergency and Sensor addresses for a given SD90 can be set to different values, there´s no particular reason for doing so and the full functionality of the SD90 can be achieved by setting both Emergency and Sensor address to the same value from 2 through 15.

Instructions for changing the Emergency and Sensor addresses are provided in the Marmitek SD90 Advanced Use manual, which at the time of this writing is available for download from URL:
http://www.marmitek.com/nl/manual/9652_SD90_AdvancedUse.pdf
however there´s currently no reference to this anywhere on the Marmitek website. The instructions are reproduced here.

Having decided on the Emergency and Sensor addresses to use, perform the following steps:

1. Press and hold the Test button, and while doing so, press and hold the Reset button until the yellow LED lights up. Then release the Reset button.
2. Release the Test button and wait 3 seconds.
3. Briefly press the Test button the number of times for the Emergency address, e.g., once for address 1, twice for address 2, etc. The LED will blink for each press.
4. Wait until the LED lights up again, then wait another 3 seconds.
5. Briefly press the Test button the number of times for the Sensor address. Then wait.

After a delay of about 3 seconds, the LED will flash the Emergency address and after another few seconds will flash the Sensor address. If the Sensor address is anything other than 1, the LED will then flash rapidly a number of times to indicate the procedure has been completed.

The programmed addresses can be recovered by pressing the Reset button. The LED will flash the Emergency address, then after a short delay the Sensor address.

The Heyu SD90 model allows the Emergency and Sensor signals to be mapped to the same or different housecode|unit addresses, depending on whether one or both IDs are supplied as a parameter in the ALIAS directive.
Examples:

```
 ALIAS Both_ID  F1  SD90  0xCA  0xDA
-- or --
 ALIAS Emer_ID  F1  SD90  0xCA
 ALIAS Sens_ID  F2  SD90  0xDA
```

(where 0x54CA and 0x54DA should be substituted for 0xCA and 0xDA respectively in the above when

using an RFXCOM receive).

The signal from the Emergency address appears in the Heyu monitor as "Test" when either the Test button is pressed or when the detector is actuated by smoke. (The SD90 makes no distinction between the two.)

The signals from the Sensor address are "Alert" when the detector is actuated by smoke, and "Clear" when the smoke dissipates and also at the heartbeat intervals.

When the SD90 determines that a low battery condition exists, it sends a single Sensor signal with the LoBat flag, then stops sending the heartbeat signal. (The detector will start issuing audible beeps at intervals.)

SPECIAL BWR102 SETUP

Mapping the BWR102 scale data to a housecode|unit address with an ALIAS directive and module type ORE_WGT1 is similar to that for other Oregon sensors.

For each weight measurement, the BWR102 retransmits the encoded weight data at intervals of 10 or 11 seconds, up to 7 times or until another weight measurement is started. The first of these transmissions will always have the ´changed´ flag set, even if the weight is identical to the previous weight measurement. Subsequent retransmissions will have this flag unset.

The weight units slider switch on the scale controls only the unit displayed on the scale´s LCD; the transmitted native units are always kilograms, to 0.1 kg precision. The configuration directive ORE_WGTSCALE is used to convert the native units to the user´s preferred units.
Example:

```
ORE_WGTSCALE  Lbs  2.200
```

AUTHORS

Charles W. Sullivan (cwsulliv01@heyu.org)

SEE ALSO

http://www.heyu.org
heyu(1), x10config(5), x10sched(5), x10scripts(5), x10cm17a(5), x10rfxsensors(5), x10rfxmeters(5), x10digimax(5), x10oregon(5)